A NUMERICAL TECHNIQUE FOR THE DIAGONALIZATION OF A SYMMETRIC MATRIX

PETER D. KJELDGAARD

A NUMERICAL TECHNIQUE FOR THE
DIAGONALIZATION OF A SYMMETRIC MATRIX

* * * * *

Peter D. Kjeldgaard

A NUMERICAL TECHNIQUE FOR THE DIAGONALIZATION OF A SYMMETRIC MATRIX

by

Peter D. Kjeldgaard

Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

United Stated Naval Postgraduate School
Monterey, California

1 9 5 8

A NUMERICAL TECHNIQUE FOR THE

DIAGONALIZATION OF A SYMMETRIC MATRIX

by

Peter D. Kjeldgaard

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

from the

United States Naval Postgraduate School

ABSTRACT

In one method of solution to the problem of making up a schedule of work in a "job-shop" type industry there arises a symmetric matrix---the conflicts among the jobs to be done. This thesis is primarily concerned with the transformation of this "conflict" matrix to another matrix---a diagonalized symmetric matrix. That is a matrix having its nonzero elements bunched or clustered near the principal diagonal. The transformation is considered as the product of simple transformations each of which interchange rows or columns of the undiagonalized or partially diagonalized matrix.

A numerical technique is devised for the systemmatic reduction of an undiagonalized symmetric matrix to successively better diagonalized matrices. The technique is programmed for use on the NCR 102A Digital Computer.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION TO THE PROBLEM UNDER CONSIDERATION

1. Introduction

In a "job-shop" type industry there are many jobs---job orders--- to be done, each requiring the use of certain of the varied facilities, special tools, and specially trained technicians available in the industry. The question of how to make up the time schedule of work to be done arises. That is which jobs should be time scheduled first considering the complexity and use of varied facilities, etc., in the industry. Then having time scheduled the first one, which job should be time scheduled next, and so forth.

The mathematical problem to be considered in this thesis will assist in the determination of an orderly procedure for time scheduling the jobs to be done and will provide the order in which jobs should be considered for time scheduling.

2. The Mathematical Problem

Given a symmetric matrix A, consisting of zero and nonzero elements; find a symmetric matrix C whose elements satisfy the following:

i) The elements of a given column of A become the elements of some column of C.

ii) The elements of a given row of A become the elements of some rows of C.

iii) The nonzero elements of the matrix C are located along the principal diagonal of the matrix. That is to say there is a clustering of elements near the principal diagonal.

iv) In addition be able to enumerate the correspondence of columns of the matrix A with columns of the matrix C.

The conditions i and ii together are equivalent to the following:

i') Columns may be rearranged, maintaining the order of elements.

ii') Rows may be rearranged, maintaining the order of elements. For consider any element of the matrix say (i, j). The column i is shifted to say the h-th column not necessarily retaining the order of elements. Also row j is shifted to say the k-th row not necessarily retaining the order of elements. The element that was in position (i, j) must now be in position (h, k). However, the same result is obtained by shifting the i-th column to the h-th column maintaining the order of elements, and then shifting the j-th row to the k-th row maintaining the order of elements. This must be true for all (i,j). Thus i and ii imply i' and ii'. That i' and ii' imply i and ii is obvious.

An assumption is made that the number of nonzero elements in any row or column is small compared to the order of the matrix, say one tenth the elements are nonzero. If this is not the case the idea of "diagonalization" or clustering the elements along the principal diagonal has little significance.

The transformation or sequence of transformation used to find the matrix C should be applicable to matrices of large order say 62 x 62 or even as large as 500 x 500. Since we are considering large order matrices it seems logical that the transformation or sequences of transformation should be susceptible to programming on a computer, say the NCR 102A Digital Computer.

3. Notation.

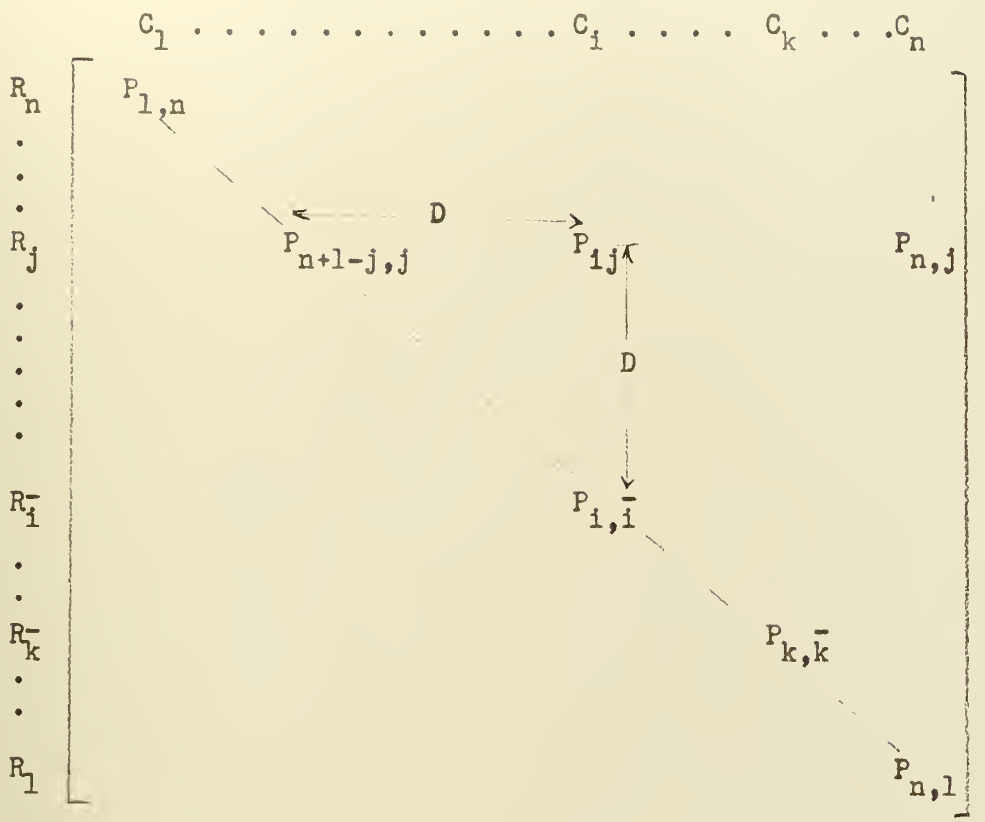


Figure 1 General Form of Matrix

A symmetric matrix A of order n, an n x n array of zero and nonzero elements. The exact value of the nonzero elements is not important--the reasons for this are explained in Chapter IV.

All nonzero elements P_{ij} are located by coordinates (C_i, R_j) indicating its position in the matrix array is in the C_ith column and the R_jth row. All elements not designated and enumerated are assumed zero.

The columns are designated $C_1, C_2, \ldots\ldots\ldots, C_n$ numbered in the usual fashion from left to right.

The rows are designated $R_1, R_2, \ldots\ldots\ldots R_n$ numbered from the bottom upward. This unusual numbering is associated with the numerical technique used to "diagonalize" the matrix.

The sum of the row and column positions is designated S_{ij}, where $S_{ij} = C_i + R_j$. Note that all the elements of the principal diagonal have the same value, namely, $n + 1$.

The distance, D, of an element from the principal diagonal is $\left| S_{ij} - (n + 1) \right| = D$. This is an integer distance along the ith column or the jth row.

4. The Measure of Diagonalization of a Matrix

Some criteria must be set up to measure the clustering or bunching of the nonzero elements along the principal diagonal to distinguish between the matrices. That is to determine which of two matrices is more diagonalized. This measure is relative and serves only as an index.

A simple criterion would be to minimize the distance of the element farthest from the principal diagonal.

$$\min_{P_{ij}} \left[\text{Max} \left| (n + 1) - S_{ij} \right| \right] \qquad \text{----Min Max D}$$

This will produce a certain amount of diagonalization . However, a simple example will show that a more powerful criterion is needed. Figures 2a and 2b both have the same Min Max D, namely 2, however, Figure 2b is more compact along the principal diagonal.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

(a) (b)

Figure 2 Examples of Min Max D Criteria

Consider the criterion that the sum of the distance from each element to the principal diagonal squared be minimized.

$$\operatorname{Min} \sum_{P_{ij}} \left[(n\ 1) - S_{ij}\right]^2 \qquad \text{---Min } \sum D^2$$

Each element far from the principal diagonal will contribute greatly to the summation, whereas elements next to the principal diagonal will have negligible effect on the summation. Thus the greater the clustering along the principal diagonal, the smaller will be the summation. The previously considered Min Max D criterion will tend to be satisfied also. For consider Figure 3 showing two columns (all other columns are assumed the same in both cases and, therefore, contribute the same to the summation in both cases).

Pr. Diagonal

$$\begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ \cdots & \\ \cdots & \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ \cdots & \end{array} \quad k$$

(a)

Pr. Diagonal

$$\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ \cdots & \\ \cdots & \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ \cdots & \end{array} \quad k$$

(b)

Figure 3 Example of $\sum D^2$ for Row Interchange

In the summation note that only the contribution of the outer element in each column is affected by the interchange of the columns under consideration in Figure 3. In (a) the outer elements contribute to the summation $(k+1)^2 + (k-1)^2$ or $2k^2+2$. In (b) these same elements contribute $2k^2$. Thus (b) is better diagonalized in the sense of both Min Max D and Min $\sum D^2$.

For the remainder of the discussion and in testing the merits of a transformation under consideration we will use Min $\sum D_2$.

CHAPTER II

SOME FEATURES OF MATRICES AND ELEMENTARY TRANSFORMATIONS

A square matrix A whose elements satisfy the conditions

$$a_{ij} = a_{ji}$$

possesses symmetry about its principal diagonal, and is called a symmetric matrix.

If $C = B\,A$, then the matrix A is said to be linearly transformed into the matrix C. The matrix B, which affects the transformation of A, may sometimes be advantageously thought of as the multiple product of very simple matrices.

$$B = B_n\, B_{n-1} \ldots\ldots B_2\, B_1$$

The transformation of A is thus accomplished through a succession of simpler transformations, the first of which is B_1, the second B_2, and so forth.

The matrix can be transformed through postmultiplication as well as by premultiplication, i.e., $C = A\,B'$. In which case the matrix affecting the transformation is decomposed into components

$$B' = B_1'\, B_2' \ldots\ldots\ldots B_n'$$

Where B_1' indicates the first simple transformation, B_2' the second, and so forth.

It is of interest to determine the simplest fundamental forms into which can arbitrary nonsingular matrix B may be decomposed, and to interpret the individual transformations which they effect upon the form

of the matrix A. These are called elementary transformations and the matrices which produce them are called elementary transformation matrices. There are three fundamental types of elementary transformations and correspondingly three fundamental types of transformation matrices. The first type amounts to an interchange of two rows or columns of the matrix A. The second type is the addition of the elements of a row or column to the corresponding elements of another row or column. The third type is the multiplication of any row or column of A by an arbitrary nonzero factor. For our purpose it is necessary to develop only type one to a greater extent.

The transformation may be affected by multiplying A by a transformation matrix. If the desired transformation is intended to affect the ROWS, A is PREmultiplied by the transformation matrix; if the COLUMNS are to be affected, A is POSTmultiplied by the transformation matrix.

The elementary transformation matrix is formed from the unit matrix by performing on it the same elementary transformation which the desired elementary transformation matrix is intended to effect in the matrix A by means of (pre-)postmultiplication. Thus I (the unit matrix) with its pth and qth rows (columns) interchanged yields a transformation matrix which by means of pre- (post-)multiplication interchanges the pth and qth rows (columns) of the matrix A. Apparently the transformation matrix has two forms depending on whether it is intended to effect a transformation of the rows or columns of A, however, it is to be noted that one form is the transpose of the other, and further for type one the two forms are identical. We will denote this elementary transformation matrix as T_{p-q} where the subscripts indicate the rows (columns) being interchanged.

As an example, for a fourth order matrix to interchange the second and third rows (columns).

$$\underset{\text{Rows}}{T_{2-3}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \underset{\text{Columns}}{T_{2-3}} = T_{23} = T_{32}$$

Consider a symmetric matrix A, pre- and postmultiplied by the elementary matrix T_{i-j}. The resulting matrix

$$A' = T_{i-j} A \, T_{i-j}$$

is a symmetric matrix with both the ith and jth rows and columns interchanged. Continuing this process we may by the elementary matrix T_{k-m} produce the matrix

$$A'' = T_{k-m} A' \, T_{k-m} = T_{k-m} \, T_{i-j} \, A \, T_{i-j} \, T_{k-m}$$

Thus by a multiple product of elementary transformation matrices pre- and postmultiplying the matrix A, the matrix A is linearly transformed into a symmetric matrix C.

The matrix C which may be obtained from the matrix A by means of a finite number of elementary transformations is said to be equivalent to A. The equivalence of matrices is a mutual relationship since the transformations are nonsingular and hence reversible. Thus if C can be obtained from A by a succession of elementary transformations, it follows that A can be obtained by means of elementary transformations from C. The state of equivalence of the two matrices A and C may be stated

$$P\,A\,Q = C \qquad \text{where P and Q are nonsingular matrices.}$$

Now if A is a symmetric matrix and $Q = Q_1Q_2 \ldots\ldots Q_n$, where each Q_i is an elementary transformation matrix T, if further

$$P = Q^T = (Q_1Q_2 \ldots Q_n)^T = Q_n \ldots\ldots Q_2Q_1$$

then C is also a symmetric matrix. Since

$$C = Q^T\,A\,Q$$

The transpose of C is then

$$C^T = Q^T\,A^T\,(Q^T)^T = Q^T\,A\,Q.$$

A matrix which is its own transpose is symmetric. So C is symmetric.

The preceding discussion has covered the elementary transformation matrices in general. Now to apply these concepts in the notation to be used in the numerical technique to be developed in Chapter III.

In order to have a symmetric matrix after premultiplication by an elementary transformation matrix T_{i-j} which interchanged the ith and jth columns we must multiply by an appropriate row interchanging elementary matrix transformation. Since the rows are designated from the bottom up, we shall designate the appropriate postmultiplying elementary matrix as $T_{\bar{i}-\bar{j}}$. Where

$$i + \bar{i} = (n + 1)$$
$$j + \bar{j} = (n + 1)$$

Examination of Figure 1 will show that this indeed is the appropriate row interchanging matrix.

Thus $A_1 = (T_{ij}\ A)\ T_{\bar{i}\bar{j}}$.

As an example consider the matrix A interchanging first the 2nd and 5th columns, then the corresponding rows.

$$A = \begin{bmatrix} 1&0&1&0&1 \\ 0&1&1&0&1 \\ 1&1&1&1&0 \\ 0&0&1&1&0 \\ 1&1&0&0&1 \end{bmatrix} \quad A' = \begin{bmatrix} 1&1&1&0&0 \\ 0&1&1&0&1 \\ 1&0&1&1&1 \\ 0&0&1&1&0 \\ 1&1&0&0&1 \end{bmatrix} \quad A_1 = \begin{bmatrix} 1&1&1&0&0 \\ 1&1&0&0&1 \\ 1&0&1&1&1 \\ 0&0&1&1&0 \\ 0&1&1&0&1 \end{bmatrix}$$

$$A' = T_{25}\ A$$

then $\bar{i} = 6 - 2 = 4$

$\bar{j} = 6 - 5 = 1$

$$A_1 = A'\ T_{41} = A'\ T_{14}$$

$$A_1 = (\ T_{25}\ A\ )\ T_{41}$$

Similarly it can be seen that if the postmultiplication elementary transformation matrix is performed first then followed by the appropriate premultiplication elementary transformation matrix we have

$$A_2 = T_{\bar{i}\bar{j}}\ (\ A\ T_{ij}\ )$$

As an example consider the matrix A interchanging first the 2nd and 5th rows, then the corresponding columns.

$$A = \begin{bmatrix} 1&0&1&0&1 \\ 0&1&1&0&1 \\ 1&1&1&1&0 \\ 0&0&1&1&0 \\ 1&1&0&0&1 \end{bmatrix} \quad A' = \begin{bmatrix} 0&0&1&1&0 \\ 0&1&1&0&1 \\ 1&1&1&1&0 \\ 1&0&1&0&1 \\ 1&1&0&0&1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1&0&1&0&0 \\ 0&1&1&0&1 \\ 1&1&1&1&0 \\ 0&0&1&1&1 \\ 0&1&0&1&1 \end{bmatrix}$$

$$A' = A\,T_{25}$$

$$A_2 = T_{41}\,A' = T_{14}\,A'$$

$$A_2 = T_{41}\,(A\,T_{25})$$

Note that T_{25} applied as a premultiplication matrix and as a post-multiplication matrix produces different transformed matrices. That is,

$$T_{41}\,(A\,T_{25}) = A_2 \neq A_1 = (T_{25}\,A)\,T_{41}$$

Since we are discussing transformations of symmetric matrices only, necessarily these elementary matrix transformations will occur in groups of two. So for convenience define a pair of elementary matrix transformations as the transformations T_{ij} applied to the rows(columns) and the corresponding $T_{\overline{ij}}$ applied to the columns (rows).

The given matrix A has a certain $\sum D^2$, say $(\sum D^2)_0$, which in the previous example was 62. If we apply the transformations

$$A_1 = (T_{ij}\,A)\,T_{\overline{ij}}$$

the transformed matrix will have a new $\sum D^2$, which for the previous example is 38. If on the other hand we apply the transformations

$$A_2 = T_{\overline{ij}}\,(A\,T_{ij})$$

a different transformed matrix is obtained with a certain $\sum D^2$ which for the previous example is 32. Thus it can be seen that different pairs of elementary transformation matrices will produce transformed matrices "diagonalized" to a greater or lesser degree. Some pairs of elementary matrix transformations may even produce a transformed matrix which is less diagonalized in the sense of Min $\sum D^2$.

CHAPTER III

PROCEDURE

1. General

Basically the procedure is to apply a pair of elementary transformation matrices to the undiagonalized or partially diagonalized matrix and test the diagonalization in the sense of minimizing $\sum D^2$.

a. If the transformed matrix is better diagonalized, then use this matrix as the matrix to be diagonalized and repeat the process applying elementary transformation matrices to this partially diagonalized matrix.

b. If the transformed matrix is NOT better diagonalized, apply a different pair of elementary transformation matrices to the undiagonalized matrix and test for $\sum D^2$ again.

We can choose the pair of elementary transformation matrices either by some random process or systematically. This thesis is concerned with a systematic procedure for improving the diagonalization of a given matrix.

2. A systematic Procedure

An element, let us designate it P_{IJ}, which lies farthest from the principal diagonal individually contributes the most to $\sum D^2$ of any element P_{ij} in the matrix. Since by symmetry there are two such elements let us consider the one with the largest S, that is S_{IJ}. It appears that a pair of elementary matrix transformations which involve this element will be a good transformation to hypothesize making. Furthermore, to move this element to a position one unit from the principal diagonal will reduce the contribution to $\sum D^2$ of this element to a minimum. However, other elements of the rows and columns interchanged will change distance

from the principal diagonal so that one must test this hypothesized transformation to see if it improves the diagonalization. If it does we make the hypothesized transformation and repeat the process again choosing the new P_{IJ} with a maximum S_{IJ}.

If the previously hypothesized transformation does not improve the diagonalization we hypothesize another transformation. If the previous elementary transformation pair T_{iJ} was a premultiplication first, i.e., row interchange first, we now hypothesize T_{Ij}, making a column interchange first such that P_{IJ} is moved to one unit from the principal diagonal. If this results in a better diagonalized matrix we make this hypothesized transformation and repeat the process again, choosing the new P_{IJ} with a maximum S_{IJ}.

If the hypothesized transformation moving P_{IJ} to a position one unit from the principal diagonal does not improve the diagonalization of the matrix, hypothesize moving P_{IJ} to two units from the principal diagonal. If neither of these produce a better diagonalized matrix, hypothesize moving P_{IJ} to three units from the principal diagonal and continue until the interchange is with the row (column) adjacent to the element P_{IJ}.

If all these hypothesized transformations with this element P_{IJ} fail to produce a better diagonalized matrix disregard this element and choose from among the remaining elements the element with the largest S_{ij}. Repeat the process of hypothesizing transformations with this element until one is found which improves the diagonalization. If none is found among all the elements with S_{ij} greater than n + 1, i.e., we have looked at all possible moves of all elements on one half of the symmetric matrix, then the matrix

which we are attempting to diagonalize is itself the best possible diagonalization in the sense of minimizing $\sum D_2$.

By hypothesizing moving this "wildest" element to one unit from the principal diagonal first, we are maximizing the possible improvement in one stage of diagonalization. The actual improvement achieved in one stage will usually be somewhat less however. The distance between the rows (columns) being interchanged must be variable since it can be shown that a system which hypothesizés making the interchange always with an adjacent row (column) will not achieve maximum diagonalization. Figure 4a is the best diagonalization possible using a hypothesis rule of always making the interchanges with the adjacent column (row). Figure 4b is the best possible diagonalization using the systematic procedure outlined above on the same matrix.

$$\begin{bmatrix} 1&0&1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&1&0&0&0 \\ 1&0&1&1&1&1&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0 \\ 0&0&1&1&1&1&0&1&0&0 \\ 0&1&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&1&0&1&1&1 \\ 0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&1&1&1 \end{bmatrix} \qquad \begin{bmatrix} 1&1&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0 \\ 0&1&1&1&1&1&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&1&1&1&1&0 \\ 0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1 \end{bmatrix}$$

a   b

Figure 4 Example of Systemmatic Procedure For Better Diagonalization

3. Systematic Procedure Applied to the NCR 102A Computer

All column and row numbering is done in the octal number system for the simple reason that the NCR 102A computes in binary number system using data inserted in the octal number system. This

OVERALL FLOW CHART-DIAGONALIZATION PROCESS

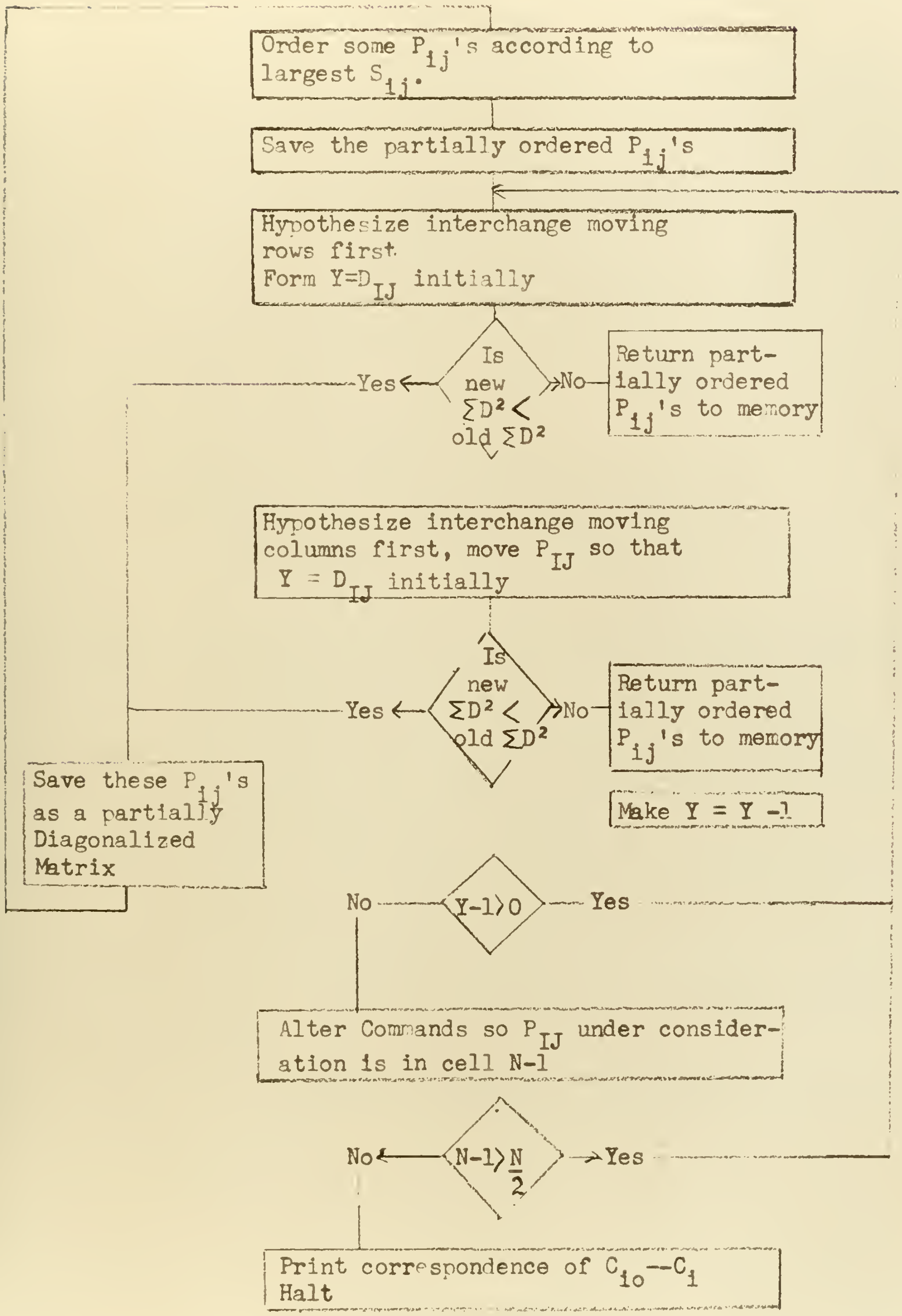
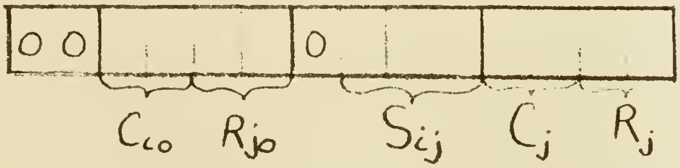


Figure 5

information could be inserted as decimal numbers. However, before the computer can start to compute a subroutine would have to be employed to convert all data to octal.

The form and amount of information about a particular nonzero element of the matrix A which will be placed in the computer is dictated in part by the computer and in part a desire by the writer to contain in one computer storage word all information relative to this particular element. Since a computer storage word is composed of 12 octal digits it is convenient to store the information about a particular P_{ij} as follows:

i) for matrices of order 63 x 63 or less



Since numbers as large as 63 can be represented in 2 octal numbers (63 decimal is 77 octal) this representation is possible. Where

C_{io} Column designation of the nonzero position P_{ij} in the matrix before any diagonalization has been effected.

R_{jo} Row designation of the nonzero position P_{ij} in the matrix before any diagonalization has been effected.

C_i Current column designation of this P_{ij} is the matrix at any stage of diagonalization.

R_j Current row designation of this P_{ij} in the matrix at any stage of diagonalization.

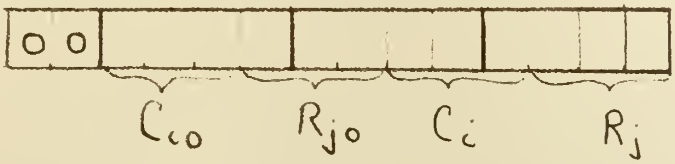
S_{ij} The current $C_i + R_j$.

It is to be noted that the C_i, R_j, and S_{ij} may change from time to time during the diagonalization process. However, the C_{io} and R_{jo} remain

unchanged and thus a one to one correspondence between the rows (columns) of the original matrix and the current partially or completely diagonalized matrix is obtainable by comparing the R_{jo} and R_j (C_{io} and C_i).

The detailed numerical techniques described later are based on stored word structures for order 63 x 63 or less.

ii) for matrices of order 511 x 511 or less



Numbers as large as 511 can be represented in 3 octal numbers (511 decimal) is 777 octal). However, certain information must be sacrificed in order to contain the essential information in one computer storage word length. For these larger matrices the value of S_{ij} must be formed wherever needed rather than being immediately available. The symbols have the same meaning as above.

Determination of the Largest S_{IJ}.

In order to determine the "wildest" element or more precisely the element with the largest S_{ij}, it is necessary to order the elements P_{ij} or at least order some of them, say the largest 6. The reason that more than one largest S_{ij} must be obtained is that it may happen that none of the hypothesized transformations using the P_{IJ} element with the largest S_{IJ} will improve the diagonalization. However, it is unlikely that among the elements with the 6 largest S_{ij}'s at least one will not have a transformation which will improve the diagonalization. If one of them does improve the diagonalization the resulting P_{ij}'s will be reordered

FLOW CHART - ORDERING S's

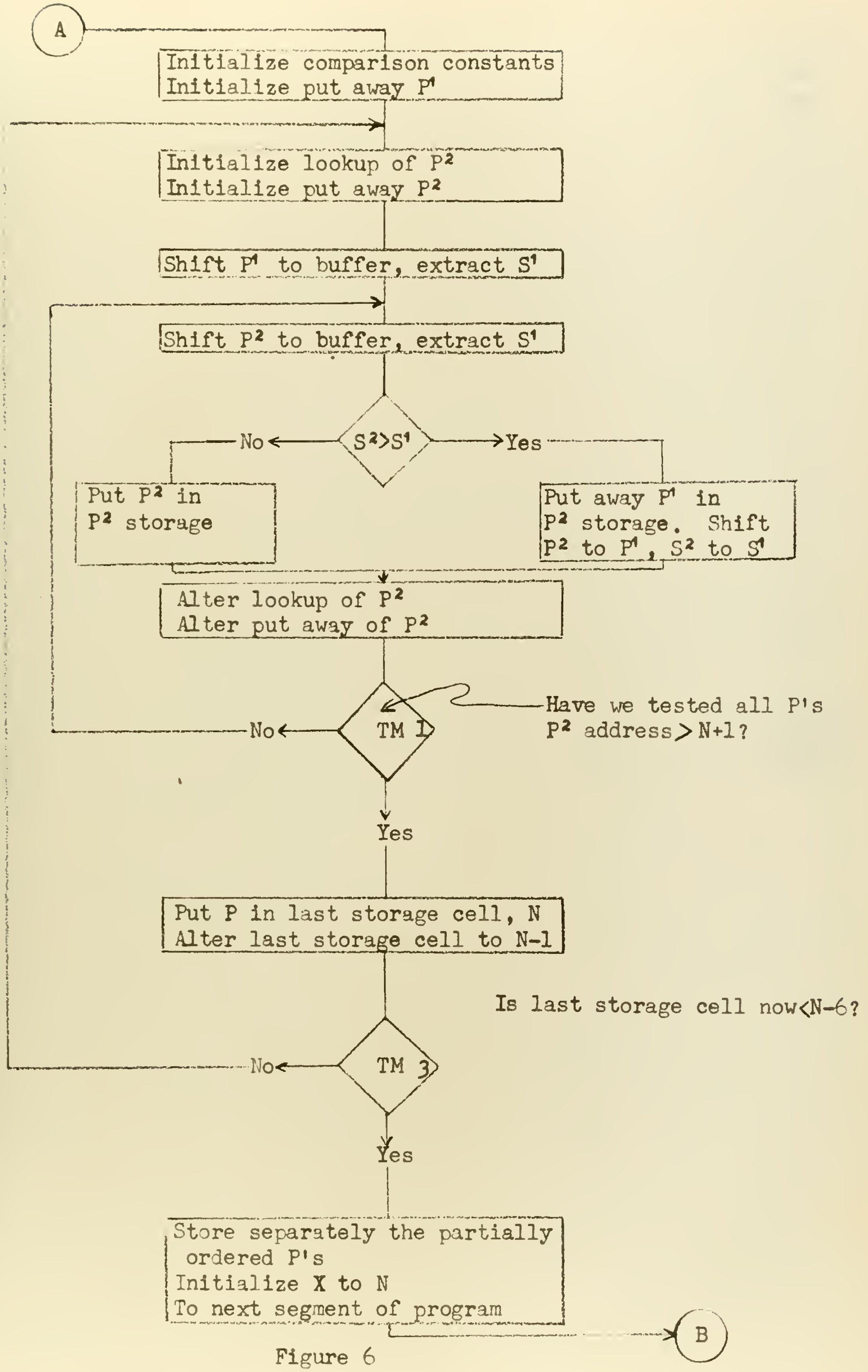


Figure 6

and the 6 largest S_{ij}'s of the new matrix obtained. It is to be noted that the technique of ordering tends to order not only the 6 largest but to relatively order all the elements. Thus after several partially diagonalized matrices are obtained, all the P_{ij}'s will be relatively well ordered.

The technique used is simply to compare the S_{ij} of the K and K + 1 storage cells, putting the P_{ij} with the smaller S_{ij} in the K storage cell. Then compare the S_{ij} of the K + 1 and K + 2 storage cells, putting in K + 1 the P_{ij} with the smaller S_{ij}. Continuing this process until, the N (last) storage cell is reached, then put both P_{ij}'s away. Decrease the number of storage cells under consideration by one, that is, for the second time through this ordering process only the first N-1 storage cells would have their S_{ij}'s compared. To determine if the preassigned number of largest S_{ij}'s have been obtained, test if N-6 is the last storage cell to be considered for the next time through the ordering process. If not repeat the process starting with the first two storage cells; if so go on to the next segment of the program.

Making a Hypothesized Transformation

For the first hypothesized transformation we consider interchanging the elements of two rows first. The P_{IJ} stored in the N storage cell will have the largest S_{IJ}. It is easily determined which row this P_{IJ} is in, namely the Jth row, and the distance from the principal diagonal, namely D. Then J-D is the row which has the Ith column on the principal diagonal.

Let $\qquad K = J - D + 1$

Then the interchange of the Jth and Kth rows will place P_{IJ} next to the principal diagonal. Now to maintain symmetry two columns must also be interchanged. It is easily seen that the columns

FLOW CHART - MAKING A HYPOTHESIZED TRANSFORMATION

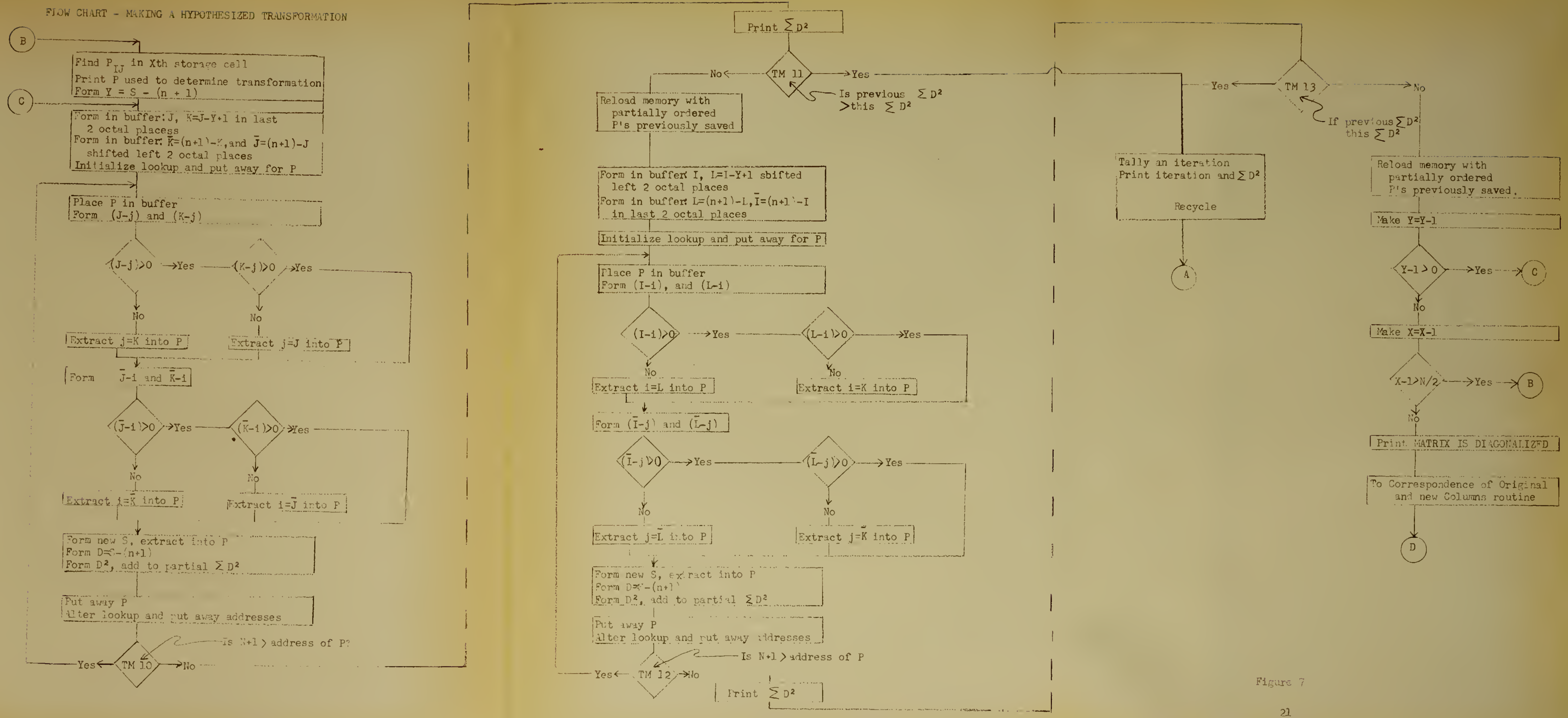


Figure 7

$$\bar{J} = (n + 1) - J$$

$$\bar{K} = (n + 1) - K$$

are the appropriate columns to interchange to maintain the symmetry. These interchanges are the equivalent of the elementary matrix transformation

$$T_{\bar{J}\bar{K}} \; A \; T_{JK}.$$

Forming these quantities J, K, $\bar{J}$, $\bar{K}$ where they are readily available proceed to examine each P_{ij} performing the following tests:

i) if $j = J$ then make $j = K$

if $j = K$ then make $j = J$

otherwise row designator remains unchanged.

ii) if $i = \bar{J}$ then make $i = \bar{K}$

if $i = \bar{K}$ then make $i = \bar{J}$

otherwise column designator remains unchanged.

Determine the new S_{ij} of this element and then D. Form D^2 and add the contribution of this element to the partial $\sum D^2$ already obtained.

When all the P_{ij}'s have been so treated, test to determine if the $\sum D^2$ for the hypothesized change is less than $\sum D^2$ for the matrix before transformation. If indeed this new $\sum D^2$ is less, we have a matrix more diagonalized than before and we start the process over again, ordering by the S_{ij}'s, etc.

If, however, the new $\sum D^2$ is not less than before discard this hypothesized transformation and replace the storage cells with the partially ordered set of P_{ij}'s which had previously been stored separately. This time hypothesize a transformation by interchanging elements of two columns first. Taking again P_{IJ} in the N storage cell determine which

column P_{IJ} is in, namely, I, and the distance from the principal diagonal. Then I - D is the column which has the Jth row on the principal diagonal.

Let $\quad L = I - D + 1$

Then the interchange of the Ith and Lth columns will place P_{IJ} next to the principal diagonal. Now to maintain symmetry two rows must also be interchanged. It is easily seen that the rows

$$\bar{I} = (n + 1) - I$$

$$\bar{L} = (n + 1) - L$$

are the appropriate rows to interchange to maintain the symmetry. These interchanges are equivalent to the elementary matrix transformation pair $T_{IL} \; A \; T_{\bar{I}\bar{L}}$.

Forming these quantities I, L, $\bar{I}$, $\bar{L}$ where they are readily available proceed to examine each P_{ij} performing the following tests:

i) if $j = \bar{I}$ then make $j = \bar{L}$

if $j = \bar{L}$ then make $j = \bar{I}$

otherwise row designator remains unchanged.

ii) if $i = I$ then make $i = L$

if $i = L$ then make $i = I$

otherwise column designator remains unchanged.

Determine the new S_{ij} of this element and then D. Form D^2 and add the contribution of this element to the partial $\sum D^2$ already obtained.

When all the P_{ij} have been so treated test to determine if the $\sum D^2$ for the hypothesized change is less than $\sum D^2$ for the matrix before transformation. If indeed this new $\sum D^2$ is less, we have a matrix more diagonalized than before and we start the process over again, ordering

by the S_{ij}'s, etc.

If, however, the new $\sum D^2$ is not less than the $\sum D^2$ for the matrix before transformation, discard the hypothesized transformation, replace the storage cells with the partially ordered set of P_{ij}'s which had previously been stored away.

Now make $K' = K + 1$

$L' = L + 1$

If $K' < J$ hypothesize making interchanges as before using K' and L' in place of K and L.

If, however, $K' = J$, it means that we have hypothesized all possible row (column) interchanges which shift the Jth row (Ith column) and none of them improve the diagonalization of the matrix. Make $N' = N - 1$, that is repeat the hypothesizing transformations process using the next wildest element. If none of these hypothesized transformations will improve the diagonalization of the matrix, make $N' = N - 2$, and so on, continuing this process until all elements have been tried. If none of the hypothesized transformations improve the diagonalization of the matrix, then the best diagonalization possible using the criteria of minimize $\sum D^2$ has been attained.

CHAPTER IV

A PHYSICAL PROBLEM

Consider a job-shop type industry such as a shipyard which may be considered typical. The products may be considered the accomplishment of job orders, each of which may require the use of special facilities, special machines and equipment, and the special talents of specialized technicians.

Let us consider several typical or probable job orders:

Job Order #1: Main Battery Range finder; clean, repaint prismatical marks on lenses, reevacuate rangefinder tube, fill evacuated tube with nitrogen, remount and realign the rangefinder.

Job Order #2: Main Battery Fire Control Equipment; correct systemic errors in synchro system between radar and optical portions of fire control system, also accomplish electronic modification XX.

Job Order #3: Binoculars; collimate and adjust 60 pairs of binoculars.

Job Order #4: Radio Equipment; overhaul model YY receiver.

And so forth....

The job orders are analyzed with respect to special machines, facilities, and specialized technical work required. For example, Job Order #1 will require, 1) all the facilities of the optical section for work space during cleaning, 2) a specialized technician to repaint the prismatical marks on the lenses, 3) special equipment to evacuate the range finder tube afterwards, 4) special equipment to refill the range finder tube with nitrogen, 5) special alignment equipment and technical assistance during remounting.

	Optical shop	Radar repair shop	Radio repair shop	sheet metal fabricating shop	. . .	Optical lense specialist	Optical alighment spec.	Synchro specialist	Aluminum welding specialist	. . .	Optical technicians	electronic technicians	. . .	Special evacuation equipment	Nitrogen filling equipment	Optical alignment tools	. . .
Job order #1	X					X	X				X			X	X	X	
Job order #2		X					X	X			X	X				X	
Job order #3	X					X					X						
Job order #4			X									X					
.																	
.																	
.																	

Figure 8 Chart of Jobs vs Facilities, Specialized Men and Equipment

Job order #2, while primarily the concern of electronics technicians and requiring electronics equipment also will require a synchro expert and technicians from the optical shop.

Job order #3 requires the shop facilities of the optical shop. Thus each job order can be analyzed and a chart prepared as indicated in Figure 8, where there are m job orders and n columnar headings.

Figure 8 in effect represents a m x n matrix of zero and nonzero elements. Each x in the figure represents a nonzero element. All the nonzero elements of a given row represent the columns (men, machines and facilities) which are affected by a particular job order. Post multiplication of this m x n matrix M with an n x m matrix, say M^T - the matrix transpose- will produce an m xm matrix A. The matrix multiplication is carried out in the usual way using the conventions

(nonzero element) x (nonzero element) = (nonzero element)

(nonzero element) x (zero element) = (zero element).

This matrix $M M^T$ is symmetric with the rows and columns representing job order numbers. Thus a nonzero element in the matrix position (i,j) indicates that job order i and job order j have some conflict between them. This conflict may be that both job orders require the use of the space facilities of the optical shop or the use of the special nitrogen filling equipment. This matrix product $M M^T$ is thus the "Job order Conflicts" matrix.

This symmetrical conflict matrix is the matrix which is diagonalized elsewhere in this thesis. The diagonalization provides an ordering of the job orders such that all job orders which conflict with a given job order are listed in the ordering near that given job order. Thus a job order which conflicts with no other job orders will appear at one end of the ordering.

Note that the ordering is not unique for the ordering may be enumerated in reverse order for the same diagonalized matrix.

Let us consider again the original job order vs men, machines and special facilities matrix M. Since there is nothing special about the order that the job orders are listed, let us rearrange the rows so that the job orders which conflict with each other are near each other. That is precisely the ordering of the job orders given by the diagonalization of the "Job Order conflict" matrix. When the matrix with the rows ordered in this fashion is scrutinized, it will be noted that there is a natural rearrangement of the columns which will yield a matrix compact along a line--- for square matrix this line is the principal diagonal and for the rectangular matrix this line runs from corner to corner. The results will be a matrix such as Figure 9.

By some criteria such as the type commanders decision that job A is most important, the first job order is selected for scheduling. Other criteria for finding the starting place are to use the job order row with the longest length or the middle row. Having selected the starting row by some criteria the complete time scheduling of job order A limits certain aspects of job orders B and C (see Figure 9). The time scheduling of B and C in turn limit D and E. This continues until a point is reached where F and G are the job orders being time scheduled. They can be scheduled independently of each other limited only by the job orders already scheduled. This is because they use none of the same men, machines or facilities. At this point the remainder of the upper half can be scheduled without concern for the lower half causing interference.

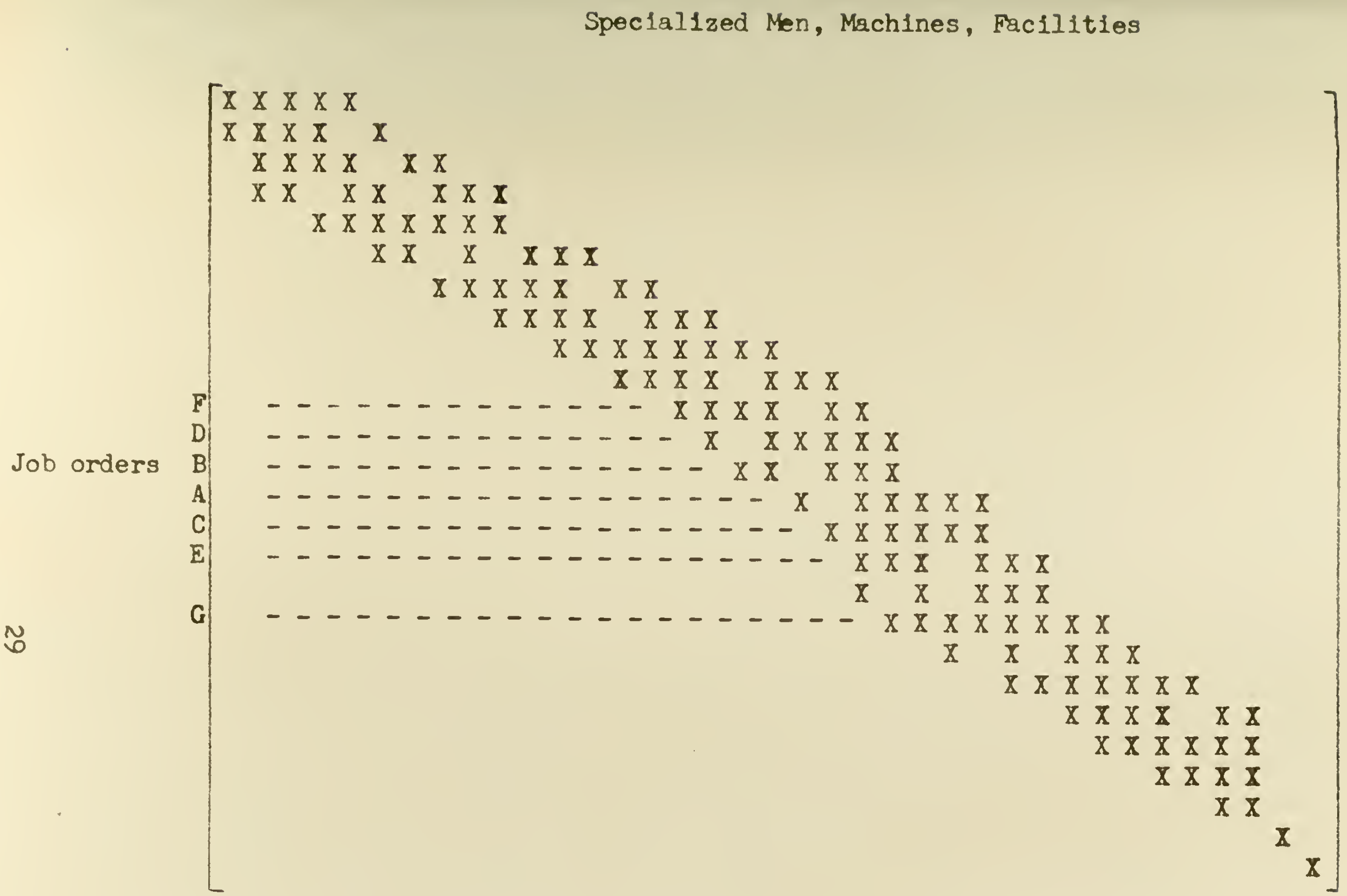


Figure 9 Rearranged Matrix

APPENDIX

The complete program is included here for reference purposes. The reader is also referred to the Flow Charts for individual segments of the program.

The program occupies cells 1200 to 1747 in the main memory. The program may be operated either minimum access or sequential numbering drum. As written the program requires also the use of a magnetic tape having blocks 500 to 620 available in both mode 1 and mode 2. The nonzero elements are stored starting in cell 0000 up to 1077. Channel 11 of the memory is reserved for temporary storage during the last section of the program--- the printout of the correspondence of columns in the undiagonalized and diagonalized matrices. If part of channel 11 is required for the storage of nonzero elements then certain cells and commands must be altered for the last segment of the program. (Constant in 1630, and command 1664 M^1 portion). These could be changed to 1200 wiping out a portion of the program no longer required. If there are less than 477 (octal) nonzero elements the program can be simply modified to eliminate the use of the tape unit. The partially ordered P's then are stored starting in 0500. The last iteration is then not saved.

The only constants which must be filled into program are cells:

1440 --- Number of last cell filled with a nonzero element

1441 --- Order of matrix plus one.

If P's are initially completely filled out start computing in 1200. If only the Co and Ro numbers are in the M^1 portion of the P's start computing in 1550. If SEN 2020 up program will type out all P's after an iteration is completed. SEN 2040 must be up for program to recycle after completing an iteration.

ADD	INST	M^1	M^2	M^3	REMARKS
					ONE TIME CONSTANT PREP
A 1200	30	1440	1460	1443	N in M^1
1201	35	1443	1454	1442	N+1 in M^1
1202	36	1440	1463	2000	Comparison const to
1203	32	2000	1444	1461	order 6 S's
1204	32	1440	1450	2100	Comparison const for
1205	35	2000	1464	2000	tape cmds
1206	32	2000	1444	1456	
					ORDERING 6 S's
1207	35	2100	2100	1477	Initialize partial ΣD^2
1210	14	0000	0100	0500	Tape block search BL500
1211	35	1442	2100	1462	Initialize X
1212	32	1440	1444	1226	Initialize put away P^1
1213	32	1612	1471	1217	Initialize lookup P^2
1214	32	2100	1615	1222	Initialize put away P^2
1215	35	0000	2100	2006	Shift P^1 to 2006
1216	32	2006	1632	2104	Extract S^1
1217	35	XXXX	2100	2007	Shift P^2 to 2007
1220	32	2007	1632	2105	Extract S^2
1221	34	2005	2004	1233	Is $S^2 > S^1$
1222	35	2007	2100	XXXX	Put away P^2
1223	35	1222	1625	1222	Alter lookup P^2
1224	35	1217	1612	1217	Alter put away P^2
1225	34	1462	1217	1217	Tested all P's?
1226	35	2006	2100	XXXX	Put away P^1
1227	36	1226	1625	1226	Alter put away add P^1
1230	36	1462	1612	1462	Reduce X to X-1
1231	34	1226	1461	1213	Ordered enough S's?
1232	34	3400	2100	1240	To next segment of program
1233	32	1222	1615	1234	Extract put away add from 1222
1234	35	2006	2100	XXXX	Put away P^1
1235	35	2007	2100	2006	Shift P^2 to P^1
1236	35	2005	2100	2004	Shift S^2 to S^1
1237	34	3100	2100	1223	

ADD	INST	M^1	M^2	M^3	REMARKS
					MAKE HYPOTHESIZED ROW FIRST TRANSFORMATION
1240	32	2100	1444	1241	Initialize Buffer load cmd
1241	05	3000	3000	XXXX	Shift partially ordered P's
1242	15	0000	0101	0000	to tape starting BL500
1243	35	1241	1467	1241	in Mode 1
1244	34	1456	1241	1241	
1245	32	1443	1471	1462	Initialize X to N
B 1246	32	1462	1471	1250	Prep lookup of P
1247	32	1462	1471	1251	Prep print cmd
1250	21	XXXX	2100	0001	Prints P
1251	35	XXXX	2100	2000	Shift P to buffer
1252	32	2000	1450	2107	Extract C to 2007
1253	32	2000	1451	2104	Extract R to 2004
1254	32	2000	1447	2101	Extract S to 2001
1255	30	2001	1453	2001	Shift S 4 octal places
1256	36	2001	1441	1457	Form Y
C 1257	36	2004	1457	2003	Form J-Y
1260	35	2003	1452	2003	Form K
1261	36	1441	2003	2005	Form $\overline{K}$
1262	30	2005	1463	2005	Shift $\overline{K}$ left 2 octal places
1263	36	1441	2004	2006	Form $\overline{J}$
1264	30	2006	1463	2006	Shift $\overline{J}$ left 2 octal places
1265	32	2100	1444	1312	Initialize put away
1266	32	2100	1471	1267	Initialize lookup
1267	35	XXXX	2100	2000	Shift P to 2000
1270	32	2000	1445	2102	Extract j to 2002
1271	36	2006	2002	2001	Form (J-j)
1272	34	2001	2100	1275	Is (J-j) greater than 0
1273	32	2005	1611	2000	If =0, j becomes K
1274	34	3400	2100	1300	
1275	36	2005	2002	2001	Form (K-j)
1276	34	2001	2100	1300	Is (K-j) greater than 0
1277	32	2006	1611	2000	If =0, j becomes J
1300	32	2000	1614	2102	Extract i to 2002
1301	36	2004	2002	2001	Form ($\overline{J}$-i)
1302	34	2001	2100	1305	Is ($\overline{J}$-i) greater than 0
1303	32	2003	1623	2000	If =0, i becomes $\overline{K}$
1304	34	3000	2100	1310	
1305	36	2003	2002	2001	Form ($\overline{K}$-i)
1306	34	2001	2100	1310	Is ($\overline{K}$-i) greater than 0
1307	32	2004	1623	2000	If =0, i becomes $\overline{J}$
1310	35	1267	1454	1267	Alter lookup for P
1311	34	3000	2100	1400	To Sum and partial D^2 S.R.
1312	35	2000	2100	XXXX	Put P away
1313	35	1312	1452	1312	Alter put away cmd
1314	34	1442	1267	1267	Operated on all P's?

ADD	INST	M^1	M^2	M^3	REMARKS
1315	21	1476	2100	0002	Print new ΣD^2
1316	21	1237	1465	0001	Tab
1317	34	1475	1477	1434	Is previous ΣD^2 greater than this ΣD^2. If greater go to 1434.
1320	34	3000	2100	1421	To subroutine Reload P's
1321	05	3000	3000	1500	Return buffer contents saved
					MAKE HYPOTHESIZED COLUMN FIRST TRANSFORMATION
1322	30	1457	1463	2006	Shift Y 2 octal places
1323	36	2007	2006	2006	Form I-Y
1324	35	2006	1464	2006	Form L
1325	30	2006	1466	2005	Shift L 2 octal places
1326	36	1441	2005	2005	Form $\bar{L}$
1327	30	2007	1466	2003	Shift I 2 octal places
1330	36	1441	2003	2003	Form $\bar{I}$
1331	32	2100	1444	1356	Initialize put away
1332	32	2100	1471	1333	Initialize lookup
1333	35	XXXX	2100	2000	Shift P to 2000
1334	32	2000	1611	2102	Extract i to 2002
1335	36	2007	2002	2001	Form (I-i)
1336	34	2001	2100	1341	Is (I-i) greater than 0
1337	32	2006	1616	2000	If =0, i becomes L
1340	34	3000	2100	1344	
1341	36	2006	2002	2001	Form (L-i)
1342	34	2001	2100	1344	Is (L-i) greater than 0
1343	32	2007	1616	2000	If =0, i becomes I
1344	32	2000	1623	2102	Extract j to 2002
1345	36	2003	2002	2001	Form $(\bar{I}$-j)
1346	34	2001	2100	1351	Is $(\bar{I}$-j) greater than 0
1347	32	2005	1626	2000	If =0, j becomes $\bar{L}$
1350	34	3000	2100	1354	
1351	36	2005	2002	2001	Form $(\bar{L}$-j)
1352	34	2001	2100	1354	Is $(\bar{L}$-j) greater than 0
1353	32	2003	1626	2000	If =0, j becomes $\bar{I}$
1354	35	1333	1612	1333	Alter lookup for P
1355	34	3000	2100	1400	To Sum and partial D^2 S.R.
1356	35	2000	2100	XXXX	Put P away
1357	35	1356	1625	1356	Alter put away cmd
1360	34	1442	1333	1333	Operated on all P's?
1361	21	1476	2100	0002	Print new ΣD^2
1362	21	1232	1465	0001	Carriage return
1363	34	1475	1477	1434	Is previous ΣD^2 greater than this ΣD^2. If greater go to 1434
1364	34	3000	2100	1421	To subroutine Reload P's

ADD	INST	M^1	M^2	M^3	REMARKS
1365	36	1457	1452	1457	Make Y=Y-1
1366	05	3000	3000	1500	Return buffer contents saved
1367	21	1237	1465	0001	Tab
1370	34	3000	2100	1371	
1371	33	1457	1452	1257	Is (Y-1) greater than 0
1372	36	1462	1454	1462	Make X=X-1
1373	21	1232	1465	0001	Carriage return
1374	30	1443	1542	1455	Form N/2
1375	33	1462	1455	1246	Is X greater than N/2?
1376	21	1532	1452	0004	Print MATRIX IS DIAGONALIZED
1377	34	3000	2100	1636	To Correspondence of C orig and C new routine
					D
					SUM AND PARTIAL D^2 SUBROUTINE
1400	30	3000	1610	2001	Prepare exit
1401	32	2001	1615	1415	" "
1402	32	2000	1614	2101	Extract R
1403	30	2001	1617	2001	Shift 2 octal places
1404	32	2000	1616	2102	Extract C
1405	35	2002	2001	2001	Add C+R
1406	30	2001	1620	2002	Shift new S LEFT 2 octal
1407	32	2002	1447	2000	Extract new S into P
1410	30	2001	1621	2001	Shift S RIGHT 2 octal
1411	36	1441	2001	2001	Form (n+1)-S=D
1412	26	2001	2001	2001	Form D^2
1413	35	1477	2001	1477	Add to partial ΣD^2
1414	37	1477	3000	1416	Test overflow
1415	34	3000	2100	XXXX	Exit to main program
1416	35	1476	1452	1476	Tally overflow
1417	35	2100	1477	1477	Wipe out overflow bit
1420	34	3000	2100	1415	
					RELOAD P's SUBROUTINE
1421	30	3000	1446	2001	Prepare exit
1422	32	2001	1444	1433	" "
1423	14	0000	0100	0500	Tape block search BL500
1424	04	3000	3000	1500	Save buffer contents
1425	32	2100	1444	1427	Prepare buffer cmd
1426	16	0000	0101	0000	Read tape
1427	04	3000	3000	XXXX	Buffer out to memory
1430	35	1427	1467	1427	Add 10 to buffer out addee
1431	34	1456	1427	1426	Have we read enough tape?
1432	35	2100	2100	1477	Clear partial ΣD^2
1433	34	3000	2100	XXXX	Exit to main program
1434	35	1473	1452	1473	Tally iteration
1435	35	1476	2100	1474	Shift partial ΣD^2
1436	35	1477	2100	1475	Shift partial ΣD^2
1437	34	3000	2100	1510	

B C

ADD	INST	M^1	M^2	M^3	REMARKS
					CONSTANTS
1440	00	0000	0000	N	Number of P's
1441	00	0000	0000	n+1	Order of matrix +1
1442	00	N+1	0000	0000	Comparison constant
1443	00	N	0000	0000	Comparison constant
1444	00	0000	0000	7777	Extractor for M^3
1445	00	0000	0000	7700	Extractor for C
1446	02	0000	0000	0030	Shift control right 8 octal
1447	00	0000	0777	0000	Extractor for S
1450	00	0000	0000	7700	Extractor for C
1451	00	0000	0000	0077	Extractor for R
1452	10	0000	0000	0001	Print control; Constant 1
1453	02	0000	0000	0014	Shift control right 4 octal
1454	10	0001	0000	0000	Print control; Constant 1
1455	00	0100	0000	0000	Print control
1456	00	3000	3000	XXXX	Comparison constant
1457	00	0000	0000	Y	Distance between rows (cols) being interchanged
1460	00	0000	0000	0030	Shift control left 8 octal
1461	00	2006	2100	XXXX	Comparison constant
1462	00	X	0000	0000	Variable constant
1463	00	0000	0000	0006	Shift control left 2 octal
1464	00	0000	0000	0100	
1465	10	0100	0000	0000	Print control
1466	02	0000	0000	0006	Shift control right 2 octal
1467	00	0000	0000	0010	
1470	00	7700	0000	0000	Extractor for Co
1471	00	7777	0000	0000	Extractor for M^1
1472	00	0077	0000	0000	Extractor for Ro
1473	00	0000	0000	0000	Tally of iterations
1474	00	0000	0000	0000	$\sum D^2$ of last iteration
1475	00	0000	0000	0000	$\sum D^2$ of last iteration
1476	00	0000	0000	0000	$\sum D^2$ of current iteration
1477	00	0000	0000	0000	$\sum D^2$ of current iteration
1500	00	0000	0000	0000	Buffer save space
1501	00	0000	0000	0000	" " "
1502	00	0000	0000	0000	" " "
1503	00	0000	0000	0000	" " "
1504	00	0000	0000	0000	" " "
1505	00	0000	0000	0000	" " "
1506	00	0000	0000	0000	" " "
1507	00	0000	0000	0000	" " "

ADD	INST	M^1	M^2	M^3	REMARKS
1510	21	1232	1465	0001	Carriage return
1511	21	1473	2100	0003	Print iteration and D^2
1512	21	1232	1465	0001	Carriage return
1513	17	2020	3000	1525	SEN 2020 UP, print P's
1514	14	0000	0100	0500	Tape block search BL500
1515	32	2100	1444	1516	Initialize buffer cmd
1516	05	3000	3000	XXXX	Buffer load
1517	15	0000	0102	0000	Tape write
1520	35	1516	1467	1516	Add 10 to buffer cmd
1521	34	1456	1516	1516	Have we written enough tape?
1522	17	2040	3000	1207	SEN 2040 UP, recycle
1523	22	0000	0000	0000	Halt
1524	34	3000	2100	1200	After halt, will recycle on compute command
1525	35	1440	1452	2001	Form N+1
1526	32	2001	1444	1527	Extract (N+1) into print cmd
1527	21	0000	2100	XXXX	Print P's
1530	21	1232	1465	0001	Carriage return
1531	34	3000	2100	1514	
1532	00	3476	4175	6360	Flexo code
1533	00	5664	6044	6447	" "
1534	00	6141	7377	6641	" "
1535	00	7160	5545	4731	" "
1536	35	1476	1452	1476	Tally an overflow
1537	35	2100	1477	1477	Wipe out overflow bit
1540	34	3000	2100	1575	
1541	00	0000	0000	0000	Buffer save space-minaccess
1542	02	0000	0000	0001	Shift control right 1 binary
1543	00	0000	0000	0000	Buffer save space-minaccess
1544	00	1633	0000	0000	Constant
1545	00	0000	0000	0000	Buffer save space-minaccess
1546	00	0000	0000	0000	
1547	00	0000	0000	0000	Buffer save space-minaccess

Only the M^1 portion, i.e., Co and Ro, need be initially put in the computer. If only the M^1 portion is filled, starting computing in cell 1550 will complete the other portions, namely, current R and C and S for all elements P. This portion of the program will also compute ar initial $\sum D^2$. If SEN 2010 up this $\sum D^2$ will be entered and the initially filled out P's saved in tape unit.

ADD	INST	M^1	M^2	M^3	REMARKS
					COMPLETION OF INFORMATION FOR P's: COMPUTE INITIAL $\sum D^2$
1550	35	2100	2100	1477	Initialize partial ΣD^2
1551	30	1440	1460	1443	Form N in M^1
1552	35	1443	1454	1442	Form N+1 in M^1
1553	32	2100	1471	1555	Initialize lookup cmd
1554	32	2100	1444	1575	Initialize put away cmd
1555	35	XXXX	2100	2000	Shift P to buffer
1556	32	2000	1470	2102	Extract Co
1557	32	2000	1472	2101	Extract Ro
1560	30	2001	1446	2001	Shift Ro 8 octal places
1561	30	2002	1446	2002	Shift Co 8 octal places
1562	35	2002	2001	2003	) Extract C and R
1563	32	2003	1444	2000	) into P
1564	30	2001	1463	2001	Shift R 2 octal places
1565	35	2002	2001	2001	Form C+R
1566	30	2001	1463	2002	Shift S 2 octal places
1567	32	2002	1447	2000	Extract S into P
1570	30	2001	1466	2001	Shift S right 2 octal
1571	36	2001	1441	2001	Form D=S-(n+1)
1572	26	2001	2001	2001	Form D^2
1573	35	1477	2001	1477	Add D^2 to partial $\sum D^2$
1574	37	1477	3000	1536	Test overflow
1575	35	2000	2100	XXXX	Put away P
1576	35	1555	1454	1555	Alter lookup
1577	35	1575	1452	1575	Alter put away
1600	34	1442	1555	1555	Have we operated on all P?
1601	21	1476	2100	0002	Print $\sum D^2$
1602	32	1440	1450	2100	Make comparison constant
1603	35	2000	1464	2000	for buffer-tape cmd
1604	32	2000	1444	1456	
1605	17	2010	3000	1434	SEN 2010 UP, shift D^2 and store P's. DO NOT HAVE SEN 2040 UP
1606	22	0000	0000	0000	
					CONSTANTS OPTIMUM SPACED
1607	00	0000	0000	0000	
1610	02	0000	0000	0030	Shift control right 8 octal
1611	00	0000	0000	7700	Extractor for C
1612	00	0001	0000	0000	Print control; const 1
1613	00	0000	0000	0002	
1614	00	0000	0000	0077	Extractor for R
1615	00	0000	0000	7777	Extractor for M^3
1616	00	0000	0000	7700	Extractor for C
1617	00	0000	0000	0006	Shift control left 2 octal

FLOW CHART - CORRESPONDENCE OF ORIGINAL AND NEW COLUMNS

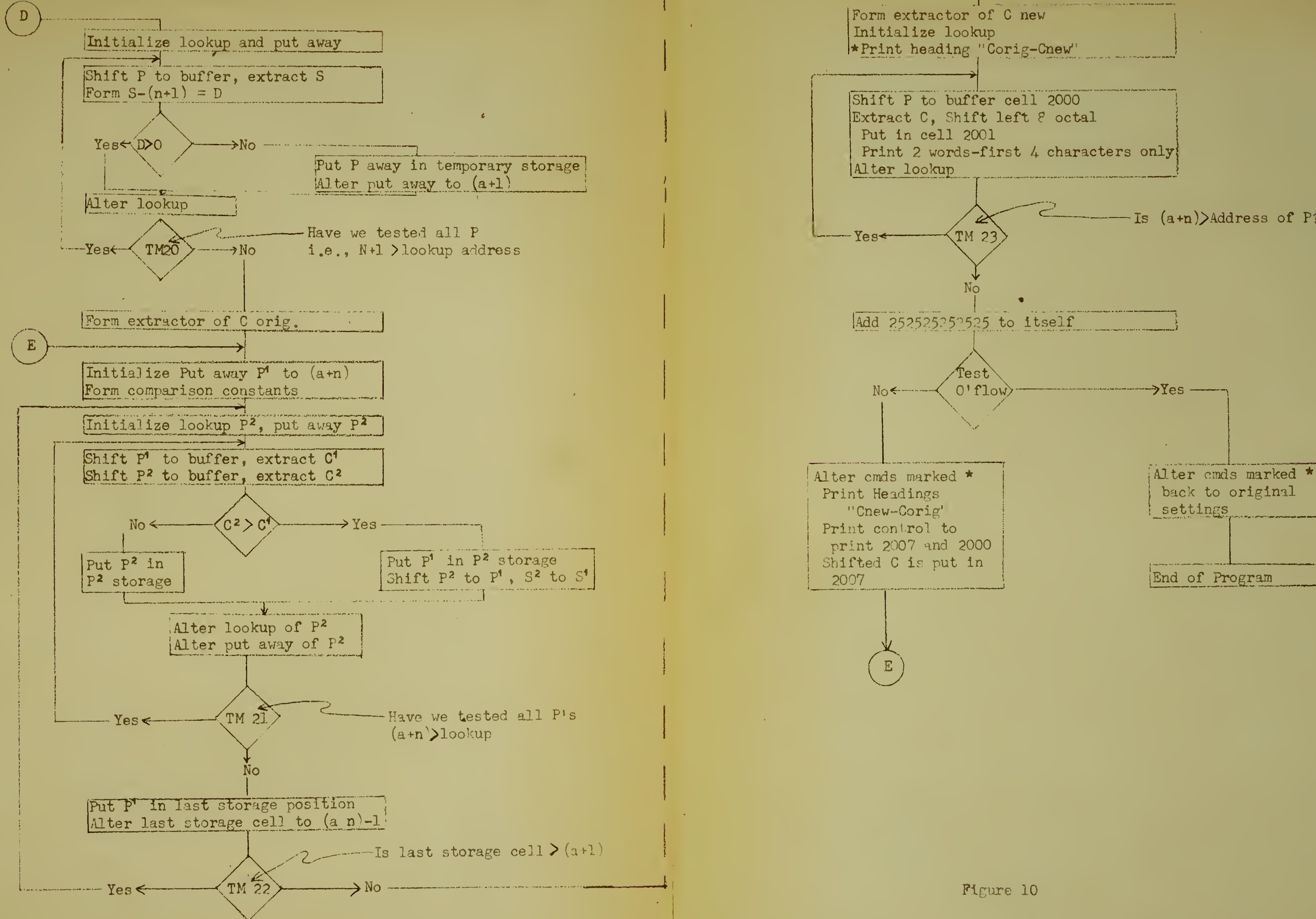


Figure 10

ADD	INST	M^1	M^2	M^3	REMARKS
1620	00	0000	0000	0006	Shift control left 2 octal
1621	00	0000	0777	0000	Extractor for S
1622	00	0000	0000	0014	Shift control left 4 octal
1623	00	0000	0000	0077	Extractor for R
1624	02	0000	0000	0006	Shift control right 2 octal
1625	00	0000	0000	0001	
1626	00	0000	0000	0077	Extractor for R
1627	00	2525	2525	2525	Test constant
1630	00	1100	0000	1100	Constant for temp storage
1631	00	1736	0000	0000	Constant
1632	00	0000	0777	0000	Extractor for S
1633	00	3436	6232	6477	Flexo code
1634	00	6360	7327	3136	" "
1635	00	6232	6466	4551	" "
D 1636	30	1441	1622	2007	Form (n+1) in M^2
1637	35	1447	2100	2004	Shift extractor of C to buffer
1640	32	2100	1471	1642	Initialize lookup
1641	32	1630	1444	1646	Initialize put away
1642	35	XXXX	2100	2000	Shift P to buffer
1643	32	2000	2004	2103	Extract S into 2003
1644	36	2003	2007	2001	Form S-(n+1)=D
1645	34	2001	2100	1650	Is D greater than 0
1646	35	2000	2100	XXXX	If =0, P is a diagonal element put in temp storage
1647	35	1646	1452	1646	Alter put away
1650	35	1642	1454	1642	Alter lookup
1651	34	1442	1642	1642	Have we tested all P's?
1652	35	1470	2100	2002	Shift extractor of Co to 2002
1653	35	1630	1441	2000	Form 1100+(n+1)
1654	36	2000	1613	2001	Form 1100+(n-1)
1655	32	2001	1444	1675	Initialize put away P^1
1656	35	2001	1452	2001	Form 1100+n
1657	30	2001	1460	2003	Shift left 8 octal
E 1660	32	1630	1444	1461	Make a comparison constant

ADD	INST	M^1	M^2	M^3	REMARKS
1661	35	1630	1454	2000	Make a comparison constant
1662	32	2000	1471	1666	Initialize lookup P^2
1663	32	1630	1444	1671	Initialize put away P^2
1664	35	1100	2100	2006	Shift P^1 to buffer
1665	32	2006	2002	2104	ExtraOt C^1 into 2004
1666	35	XXXX	2100	2007	Shift P^2 to buffer
1667	32	2007	2002	2105	Extract C^2 into 2005
1670	34	2005	2004	1702	Is $C^2 > C^1$
1671	35	2007	2100	XXXX	Put away P^2
1672	35	1666	1454	1666	Alter lookup P^2
1673	35	1671	1452	1671	Alter put away P^2
1674	34	2003	1666	1666	Tested all P's?
1675	35	2006	2100	XXXX	Put away P^1
1676	36	1675	1452	1675	Alter put away P^1
1677	36	2003	1454	2003	Alter comparifon constant
1700	34	1675	1461	1662	All P's ordered?
1701	34	3000	2100	1707	
1702	32	1671	1444	1703	Prepare put away P^1
1703	35	2006	2100	XXXX	Put P^1 away
1704	35	2007	2100	2006	Shift P^2 to P^1
1705	35	2005	2100	2004	Shift S^2 to S^1
1706	34	3000	2100	1672	
1707	30	2001	1460	2003	Prepare a comparison constant
1710	21	1633	1452	0003	Print headings
1711	21	1232	1465	0001	Carriage return
1712	35	1450	2100	2002	Shift extractor for C to 2002
1713	32	1630	1471	1714	Initialize lookup
1714	35	XXXX	2100	2000	Shift P to buffer
1715	32	2000	2002	2101	Extract C
1716	30	2001	1460	2001	Shift C left 8 octal
1717	21	2000	1455	0002	Print Co - C
1720	35	1714	1454	1714	Alter lookup
1721	21	1232	1465	0001	Carriage return
1722	34	2003	1714	1714	Have all C's been printed?
1723	35	1627	1627	1627	Make tally constant
1724	37	1627	3000	1731	Is this 2nd time thru routine?
1725	32	1631	1471	1710	Change print control
1726	32	1220	1471	1717	Change print control
1727	32	1217	1444	1716	Change C put away address
1730	34	3000	2100	1653	
1731	35	2100	1627	1627	Wipe out overflow bit
1732	32	1544	1471	1710	Change print control
1733	32	1252	1471	1717	Change print control
1734	32	1255	1444	1716	Change C put away address
1735	22	0000	0000	0000	Final halt
1736	00	3436	6232	6466	Flexo code
1737	00	4551	3136	6232	" "
1740	00	6477	6360	7327	" "

E